# Expected Value of Partial Perfect Information in Hybrid Models using Dynamic Discretization

Barbaros Yet, Anthony Constantinou, Norman Fenton, Martin Neil

***Abstract*— In decision theory models, Expected Value of Partial Perfect Information (EVPPI) is an important analysis technique that is used to identify the value of acquiring further information on individual variables. EVPPI can be used to prioritize the parts of a model that should be improved or identify the parts where acquiring additional data or expert knowledge is most beneficial. Calculating EVPPI of continuous variables is challenging, and several sampling and approximation techniques have been proposed. This paper proposes a novel approach for calculating EVPPI in Hybrid Influence Diagram (HID) models (these are Influence Diagrams (IDs) containing both discrete and continuous nodes). The proposed approach transforms the HID into a Hybrid Bayesian Network (HBN) and makes use of the Dynamic Discretization (DD) and the Junction Tree (JT) algorithms to calculate the EVPPI. This is an approximate solution (no feasible exact solution is possible generally for HIDs) but we demonstrate it accurately calculates the EVPPI values. Moreover, unlike the previously proposed simulation-based EVPPI methods, our approach eliminates the requirement of manually determining the sample size and assessing convergence. Hence, it can be used by decision-makers who do not have deep understanding of programming languages and sampling techniques. We compare our approach to the previously proposed techniques based on two case studies.**

***Index Terms*—Bayesian Networks, Dynamic Discretization, Expected Value of Partial Perfect Information, Hybrid Influence Diagrams, Value of Information.**

## I. Introduction

Value Of Information (VOI) is a powerful technique in decision analysis that identifies and prioritizes the parts of a decision model where additional information is expected to be useful. Specifically, VOI identifies the potential gain that could be acquired when the state of a currently unknown variable becomes known before the decision is made [1].

A convenient, but rather ineffective, VOI technique is called the *Expected Value of Perfect Information* (EVPI) which provides an aggregate measure showing the expected gain when we have perfect information about the states of all the variables in the model. EVPI can be easily computed with sampling methods, but its benefits are limited. A decision analyst is normally interested in the value of acquiring additional information on specific individual variables rather than an aggregate value over all variables. In such cases, the Expected Value of Partial Perfect Information (EVPPI) is used to measure the potential gain from the perfect information on individual (or subgroups) of variables. However, in contrast to the EVPI, computation of the EVPPI of continuous variables can be difficult. Several techniques (which we review in Section III) have been proposed [2-8]. These techniques were developed specifically for sampling-based modelling approaches. Although some have been implemented as R packages or online apps, they still require the user to compute the posteriors of their model by using sampling techniques. This requires technical knowledge and programming skills to undertake necessary modelling, computation and sampling to assess convergence. As a result, the use of these sampling-based techniques is limited to domains where such experts are available.

In this paper, we present a novel approach for calculating EVPPI for individual variables using an extended type of Influence Diagram (ID) and recent developments in Bayesian inference algorithms. An ID is a probabilistic graphical model that is able to represent large decision problems in a compact way and is a powerful and flexible modelling tool for decision analysis [9]. IDs that contain both discrete and continuous variables are called Hybrid IDs (HIDs). Many popular decision analysis modelling tools, including Decision Trees (DTs), Markov Models (MMs) and Bayesian decision models, can be transformed into an equivalent ID (which we discuss in Section II). Recent advances in inference algorithms make it possible to solve increasingly complex HID models efficiently [10, 11]. As a result, IDs offer a flexible and powerful modelling tool for decision analysis. Novel contributions of this paper include the following:

- It proposes an EVPPI technique that uses a completely different approach than the previous sampling-based approaches for calculating EVPPI for individual variables. Because our approach uses the Dynamic Discretization (DD) and Junction Tree (JT) algorithms. it does not require users to assess convergence of its

This work was supported by ERC project ERC-2013-AdG339182-BAYES_KNOWLEDGE. We are also indebted to Agena for use of their software.

B.Yet is with Department of Industrial Engineering, Hacettepe University, Ankara, Turkey (e-mail: barbaros.yet@hacettepe.edu.tr).

A. Constantinou, N. Fenton and M. Neil are with School of Electronic Engineering and Computer Science, Queen Mary University of London, E1 4NS, UK (e-mails: a.constantinou@qmul.ac.uk, n.fenton@qmul.ac.uk, m.neil@qmul.ac.uk).

results since this is automatically handled by the underlying algorithm. This makes the proposed method accessible to a much wider class of end users who are interested in VOI analysis.

- It proposes approximations of the proposed method to trade-off accuracy with speed. The performance of the proposed approach and its approximations are evaluated in two case studies from the health-care domain. Each case study compares our approach and its approximations with previous approaches in terms of accuracy, computation time and usability.

The paper also illustrates how different decision modelling techniques that are commonly used in the health economics domain can be represented as an equivalent HID. As a result, our EVPPI approach can be applied to a wide variety of decision problems as different modelling tools can be transformed into an HID.

The case studies also illustrate the performance of DD in solving HIDs that have mixture distributions with constants for their utility distributions. Computing the posteriors of such models is challenging as their state space is likely to have point values. The DD algorithm removes states with zero mass to prevent their exponentiation.

The paper is structured as follows: Section II provides an overview of IDs and discusses how other popular decision modelling techniques can be transformed into an ID. Section III reviews the previous methods for computing EVPPI, and shows how EVPPI is generally computed in IDs. Section IV presents our method for calculating EVPPI in HIDs. Section V illustrates its application to a case study, and Section VI presents our conclusions.

## II. Influence Diagrams (IDs)

An ID is an extension of a Bayesian Network (BN) for decision problems. In this section, we give a recap of BNs and IDs, and show how other popular modelling approaches such as DTs and MMs can be represented as an ID.

### A. Bayesian Networks (BNs)

A BN is a probabilistic model that is composed of a graphical structure and a set of parameters. The graphical structure of a BN is a Directed Acyclic Graph (DAG). Each node of the DAG represents a random variable and each directed edge represents a relation between those variables. When two nodes, *A* and *B*, are connected by a directed edge $A \rightarrow B$, we call *A* a parent and *B* a child. Each child node has a set of parameters that defines its Conditional Probability Distribution (CPD) conditioned on its parents. If both the child node and its parent nodes are discrete nodes, the CPD is encoded in a Node Probability Table (NPT). BNs that contain both discrete and continuous nodes are called Hybrid BNs (HBNs).

The graphical structure of a BN encodes conditional independence assertions between its variables. For example, a node is conditionally independent from the rest of the BN given that its parents, children and the parents of their children are observed (see Pearl [12] and Fenton and Neil [13] for more information on BNs and their conditional independence properties). The conditional independence assertions encoded in the DAG enables a BN to represent a complex joint probability distribution in a compact and factorized way. BNs have established inference algorithms that make exact and approximate inference computations by exploiting the conditional independence encoded in the structure. Popular exact algorithms, such as the JT algorithm [14], provide efficient computations for BNs with only discrete variables by transforming the BN structure into a tree structure with clusters. Exact solutions are also available for a class of HBNs in which the continuous nodes are Gaussian. While there is no feasible exact algorithm possible for computing general HBNs (i.e. without Gaussian distribution constraints), efficient and accurate approximate algorithms have recently been developed [10].

### B. Influence diagrams

An ID is an extension of BNs for decision problems [9, 15, 16]. While all nodes in a BN represent random variables, an ID has two additional types of nodes representing decisions and utilities. Thus, the types of nodes in this ID are:

- **Chance Node:** A chance node (drawn as an ellipse) is equivalent to a BN node. It represents a random variable and has parameters that define its CPD with its parents. We distinguish two classes of chance nodes in an ID:
  - Observable chance nodes, $\bar{O} = O_1, \dots, O_p$: These precede a decision and are observable at the time of, or before, the decision is made
  - Unobservable chance nodes, $\bar{N} = N_1 \dots N_q$.
- **Decision Node:** A decision node (drawn as a rectangle) represents a decision-making stage. An ID may contain multiple decision nodes, $\bar{D} = D_1, \dots, D_k$, each with finite, discrete mutually exclusive states. Each decision node $D_i$ has a set of decision states $d_{i1}, d_{i2}, \dots, d_{in_i}$ A parent of a decision node is connected to it by an 'information' arc (shown by a dashed line) representing that the state of a parent must be known before the decision is made. The information arcs in an ID only define the sequential order of the decisions $\bar{D} = D_1, \dots, D_k$ and observable chance nodes $\bar{O} = O_1, \dots, O_p$. Therefore, a decision node does not have parameters or an associated CPD.
- **Utility Node:** A utility node (drawn as a diamond) has an associated table that defines the utility values or distribution for all state combinations of its parents. There may be multiple utility nodes $\bar{U} = U_1, \dots, U_l$ in an ID, and these nodes have a child utility node that aggregate the utilities by a conditionally deterministic equation [17]. Nodes of other types cannot be a child of a utility node.

A HID is an extension of an ID in which utility nodes $\bar{U}$, and observable and unobservable chance nodes, $\bar{O}$ and $\bar{N}$, can be either discrete or continuous.

An example ID model is shown with its parameters in Figure 1. This ID models the following decision problem: *"A clinician evaluates two mutually exclusive and exhaustive diagnosis hypotheses (D). According to the clinician, the probabilities that the patient has disease X and Y are 0.25 and 0.75 respectively. Two treatment options (T), treatments A and B, are available to treat these diseases, which are effective for diseases X and Y respectively. The clinician can order a diagnostic test (S) that can decrease the uncertainty about the presence of the diseases. The probability of a positive test (R) result is 0.9 when disease X is present, and it is 0.2 when disease Y is present."*

Note that there is a sequential order between the test (*S*), the test result (*R*), and the treatment (*T*), and this order is shown by information arcs (i.e. dashed lines) in the ID. Incoming arcs to chance and utility nodes (shown by solid lines) represent CPDs or deterministic functions in the same way as a BN. The decision problem is asymmetric as the test result *(R)* cannot be observed if the test *(S)* is not made. This is modelled by adding a state named "NA" (representing "not applicable") to *R*.

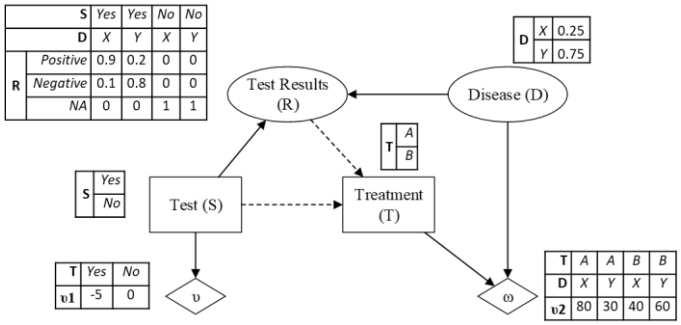


Figure 1 Influence Diagram for Treatment Selection

IDs offer a general and compact representation of decision problems. It is possible to transform other popular decision modelling approaches to IDs. In Sections II.B.1, II.B.2 and II.B.3 we describe how DTs, MMs and Bayesian decision models can be represented as IDs respectively.

*1) Decision Trees (DTs)*

A DT models a decision problem by showing all possible combinations of decision and observations in a particular sequence on a tree structure. A DT also has decision, chance and utility nodes shown by rectangle, circle and diamond shapes respectively (see Figure 2). Each outgoing arc from a decision node represents a decision alternative, and each outgoing arc from a chance node represents an outcome labelled with its name and probability. The utility nodes are located at the leaves of the tree structure and cannot have outgoing arcs. As a result, each path from the root node to a leaf node represents a decision scenario with a sequence of decisions and observations. DTs have been popular decision modelling tools due to the simplicity of their use and computation. However, the size of a DT grows exponentially as its number of variables or states increases.

There is a large literature on the use of IDs as an alternative to DTs, as IDs can represent a decision problem in a more compact way than DTs [9, 15]. Figure 2 shows the DT representation of the same decision problem as Figure 1. IDs represent each decision and chance variable with a single node; whereas in a DT a variable requires multiple associated nodes if it is conditioned on other variables in the DT. For example, *D* is modelled with a single node in the ID of Figure 1 but it needs to be modelled with four nodes in the DT of Figure 2. Moreover, adding two states to *S* would double the size of the DT shown in Figure 2 but it would not change the graphical structure of the ID in Figure 1. The additional states would only change the NPT of *S* and *R* in the ID. Therefore, it is widely accepted that IDs provide a clearer and more compact representation of complex decision problems.

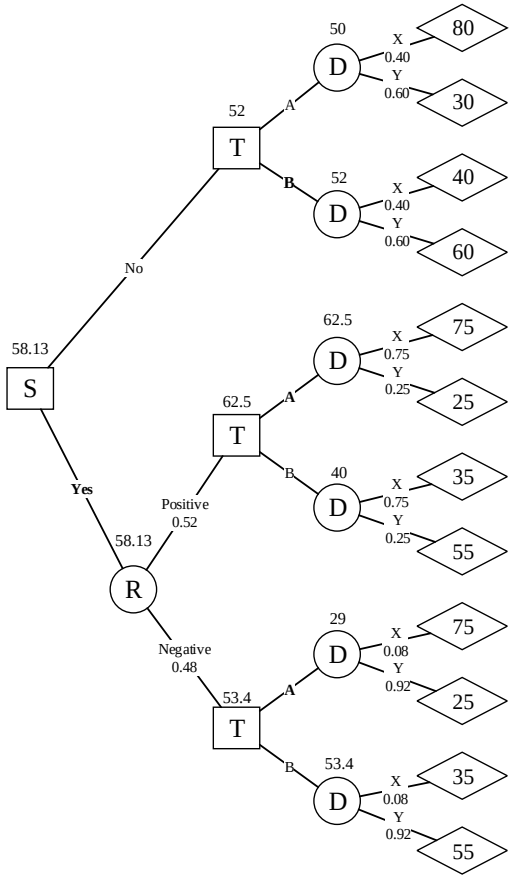


Figure 2 Decision Tree for Treatment Selection

*2) Markov Models (MMs)*

In medical decision-making, a MM is used for evaluating the outcomes of a decision over time. A MM is composed of discrete time stages. The state of the system in a time stage is only dependent on the previous time stage. A MM is called a Hidden MM (HMM) if the state cannot be entirely observed at a time stage. Both MM and HMM models can also be represented as a Dynamic BN (DBN) [18]. A DBN is an extension of BNs that has a replicated BN structure for different time stages.

Figure 3 shows a simple MM that evaluates the state of a patient over time. The patient can be 'Healthy', 'Sick' or 'Dead', and the state of a patient at a time stage only depends

on the previous time stage. Figure 4 shows a DBN representation of this model. In this example, the transition probabilities are fixed hence each node has the same NPT shown in Figure 4. The time stages in the DBN can repeated to analyze the model over a desired time period. MMs with time-dependent transition probabilities can also be modelled as DBNs that have different NPT parameters for different time stages.

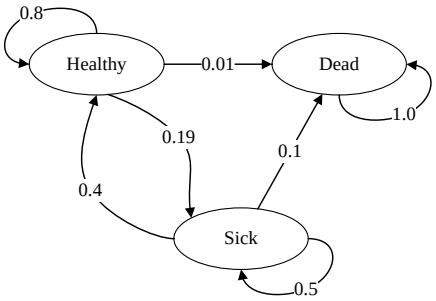


Figure 3 Markov State Transition Model


| t-1 | | Healthy | Sick | Dead |
|---|---|---|---|---|
| t | Healthy | 0.80 | 0.4 | 0 |
| | Sick | 0.19 | 0.5 | 0 |
| | Dead | 0.01 | 0.1 | 1 |

Figure 4 Dynamic Bayesian Network

In clinical decision making models, MMs of outcomes are often combined with DTs to analyze the outcome of a decision over a long period of time (see Chapter 7 of Sox, et al. [19], and Chapter 10 of Hunink, et al. [20], Fenwick, et al. [21]). It is also possible to do this by combining an ID model with a DBN. For example, Figure 5 shows a combination of the ID model in Figure 1 with the DBN model in Figure 4. This model analyses the outcomes of the treatment selection model in Figure 1 over multiple years.

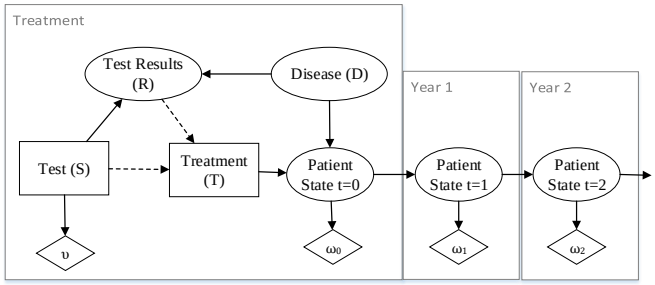


Figure 5 Influence Diagram combined with a Dynamic Bayesian Network

*3) Bayesian decision models*

Bayesian decision models are becoming increasingly popular in the health economics domain. Many EVPPI techniques [2-7] have been specifically developed for Bayesian models computed by Monte Carlo (MC) or Markov Chain Monte Carlo (MCMC) sampling approaches. These models are often represented by a set of mathematical equations that show the CPDs and the functions of each parameter.

Table 1 shows a simple Bayesian cost-effectiveness model example that aims to evaluate the net benefit of two treatment alternatives based on Response to Treatment (*RT*) and Side Effects (*SE*) of each treatment. Treatment *A* is a safer option as it has a smaller risk of leading to a side effect and the response to treatment is fairly consistent regardless of whether it is applied by experienced or trainee clinicians. Treatment *B* has better outcomes than treatment *A* when it is applied by experienced clinicians, but it also has higher risk of causing a side effect. Moreover, special clinical skills are required to apply treatment *B* and therefore the response to treatment can be a lot worse when it is applied by clinicians who are inexperienced with this treatment. Note that the uncertain parameters and utility relevant to each treatment decision are modelled separately; hence, there is not a separate decision variable in this model.

Table 1 Bayesian Cost-Effectiveness Model Example

| | |
|---|---|
| **Clinical Experience (CE)** | $P(CE = Trainee) = 0.7$ |
| | $P(CE = Experienced) = 0.3$ |
| **Side Effect (SE)** | *SE_A~ Beta(15, 75)* |
| | *SE_B ~ Beta(2,8)* |
| **Response to Treatment* (RT)** | $RT\_A_{Trainee} \sim Beta(68, 32)$ |
| | $RT\_A_{Experienced} \sim Beta(70, 30)$ |
| | $RT\_B_{Trainee} \sim Beta(2, 1)$ |
| | $RT\_B_{Experienced} \sim Beta(90, 10)$ |
| **Net Benefit (NB)** | *NB_A = 1000 × RT_A – 4000 × SE_A* |
| | *NB_B = 1000 × RT_B – 4000 × SE_B* |

An ID equivalent of this Bayesian cost-effectiveness model can be built by defining a decision node for the treatment decision being analyzed and defining utility and chance nodes corresponding to the variables in Table 1. There are usually multiple ID equivalents of a Bayesian cost-effectiveness model. For example, Figure 6 shows a compact ID that is equivalent to the model in Table 1. In this ID, *RT* and *SE* are modelled as a child of the treatment (*T*) decision. This ID structure enables us to compute the optimal decision strategy, and the EVPPI of Clinical Experience (*CE*). However, to analyze the EVPPI of *RT* or *SE* using the EVPPI technique proposed in this paper requires a different structure (which we present in Section III); this is because we are adding an information arc from the chance node analyzed to the relevant decision node and this would mean an arc from, say, *SE* to *T*, which introduces a cycle into this model. Since IDs are DAGs, the ID structure in Figure 6 therefore does not allow EVPPI analysis of *SE* or *RT*.

An equivalent ID structure that enables the EVPPI analysis on *SE* and *RT* is shown in Figure 7. In this model the parameters of *SE* and *RT* are unfolded as separate variables for each treatment. The decision node *T* is the parent of the utility node (*NB*) as it modifies the utility distribution according to SE and RT of each treatment option. Adding an information arc from *SE* and *RT* of treatments to the decision node does not introduce a cycle in this ID. Since *SE* and *RT* of each treatment option are modelled as separate nodes, this ID also allows the EVPPI analysis of *SE* or *RE* of only one treatment.

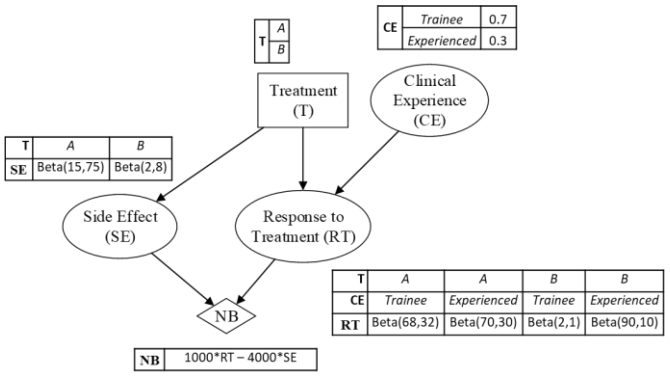


Figure 6 Equivalent Compact ID

An ID equivalent to a Bayesian model can be built if each parameter is represented separately and the decision variable is added only as a parent of the utility node to adjust the utility function for each decision, as shown in Figure 7. This approach does not lead to a compact and simple ID but it ensures that the resulting ID can make the EVPPI analyses of all variables in the corresponding Bayesian cost-effectiveness model.

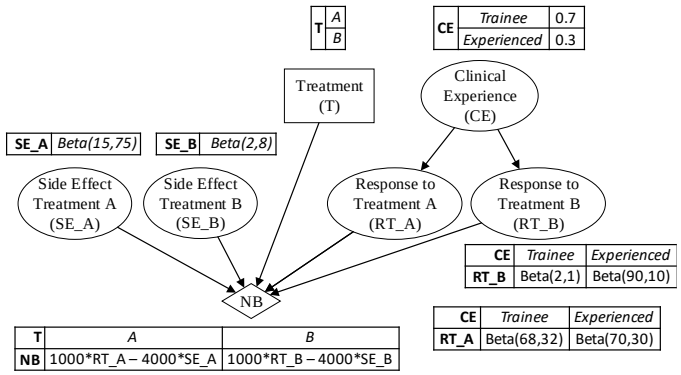


Figure 7 Equivalent ID for EVPPI analysis of SE and RT

## III. VALUE OF INFORMATION: FORMAL DEFINITIONS

Consider a decision analysis model consisting of a set of possible decision options $D$ and a set $\theta$ of uncertain parameters with the joint probability distribution $P(\theta)$. For each decision option $d \in D$ the model aims to predict the utility of $d$ denoted by $U(d,\theta)$. The expected utility of each decision option $d$ is

$$E_\theta\{U(d,\theta)\} = \sum_\theta U(d,\theta)P(\theta) \quad (1)$$

If we do not know the value of any parameter in the model, we would calculate the expected utility of each decision option and select the decision option with the maximum expected utility, i.e.

$$\max_d [E_\theta\{U(d,\theta)\}] \quad (2)$$

If we could gather perfect information on all uncertain parameters in the model, then we can change our decisions to maximize the outcome based on this information, and eliminate the losses caused by the uncertainty in the model. In this case, the expected utility with perfect information is calculated as:

$$E_\theta\left(\max_d [U(d,\theta)]\right) = \sum_\theta P(\theta) \max_d [U(d,\theta)] \quad (3)$$

The expected value of perfect information (EVPI) is the difference between the maximum expected utility with perfect information and the maximum expected utility:

$$EVPI(\theta) = E_\theta\left(\max_d [U(d,\theta)]\right) - \max_d [E_\theta\{U(d,\theta)\}] \quad (4)$$

The EVPI can be calculated by using Monte Carlo sampling techniques but it has limited use for a decision analyst who would like to know the most beneficial improvements in a model. Analysts are usually interested in the value of information of specific individual variables so that they can identify the parts of the model that are most advantageous to improve. In this case, the EVPPI for individual parameters is used. Suppose $\theta$ is divided into two subsets, the parameter $\theta^x$ and the rest of the parameters $\theta^{-x}$. If we collect perfect information about the true state of $\theta^x$, the expected net benefit given this information on $\theta^x$ is:

$$E_{\theta^x}\left(\max_d \left[E_{\theta^{-x}|\theta^x}\{U(d,\theta)\}\right]\right) \quad (5)$$

The EVPPI of $\theta^x$ is calculated as:

$$EVPI(\theta^x) = E_{\theta^x}\left(\max_d \left[E_{\theta^{-x}|\theta^x}\{U(d,\theta)\}\right]\right) - \max_d [E_\theta\{U(d,\theta)\}] \quad (6)$$

The EVPPI can be solved analytically for linear models and multi-linear models with independent inputs. Multi-linear models with correlated inputs sometimes also have analytic solutions [3, 6]. However, apart from these few special cases, complex techniques are required to calculate the EVPPI. Several techniques are available for computing EVPPI in health economics models, and most of them are developed for Monte Carlo sampling-based approaches. In the remainder of this section, we review the EVPPI techniques developed for MC Sampling based approaches (Section III.A) and examine how EVPPI is computed in IDs (Section III.B).

### A. EVPPI in Monte Carlo (MC) Sampling

Two-level MC sampling with nested outer and inner loops can be used when the EVPPI cannot be solved analytically. In this technique, the inputs are sampled in the outer loop and the remaining parameters are sampled within the inner loop. Nested MC sampling can demand excessive computation resources and time, even for moderately sized models [2]. It can also generate biased results if small sized inner samples are used [22]. The computational burden may further increase if the inputs are correlated and the conditional distributions are difficult to sample [2].

Several approximation methods have been proposed to calculate the EVPPI using one-level MC sampling. Sadatsafavi, et al. [5], Strong and Oakley [3], Strong, et al. [4] use the data generated by probabilistic sensitivity analysis to calculate an approximate EVPPI. For analysis of individual parameters, Strong and Oakley [3] partition the output data into bins and calculate an approximate EVPPI. For multiple parameters,

Strong, et al. [4] use the data to build a non-parametric regression model where the dependent variable is the net benefit and the independent variables are the parameters that are analyzed for the EVPPI. They used Generalized Additive Models (GAM) and Gaussian Process (GP) approaches, as flexible regression methods are required for estimating EVPPI. The methods of Sadatsafavi, et al. [5], Strong and Oakley [3] and Strong, et al. [4] have been implemented in the BCEA package in R [23]. Online applications for running these methods are also available but they require the results of the model to be analyzed in the form of a large set of MC or MCMC samples [24, 25]. Readers are referred to Heath, et al. [8] for a detailed review of different approximate sampling based methods for computing VOI and EVPPI [3-5, 7, 26].

### *B. EVPPI in Influence Diagrams*

EVPPI analysis in an ID examines the impact of changing the order of observations by observing one previously unobservable chance node and adding it to the sequential order. EVPPI is therefore equivalent to modifying the structure of the ID by adding an information arc and computing the difference between the modified and original IDs.

Let $G$ be an ID, and $X$ be an unobservable chance node in G i.e. $X \in \overline{N_G}$. In order to analyse EVPPI of $X$ for the decision $D \in \overline{D_G}$ let $G'$ be a modified version $G$ where an information arc is added from $X$ to $D$, and the rest of the graph is the same. As a result, $X \in \overline{O_{G'}}$ in $G'$. Note that, this information graph must not introduce a cycle as IDs are DAGs. The EVPPI of observing $X$ is the difference between the expected utilities of $G'$ and $G$.

$$EVPPI(X) = EU(G') - EU(G) \quad (7)$$

For example, Figure 8 shows the modified IDs used for computing EVPPI of knowing the state of the node *SE_B (Side Effect* of treatment *B)* in the cost-effectiveness ID shown in Figure 7. In the original ID (Figure 7), *SE_B* is unobservable and thus there is no information arc connected to it. In the modified ID (Figure 8), *SE_B* is observed before making the decision because it is connected to the decision node by an information arc.

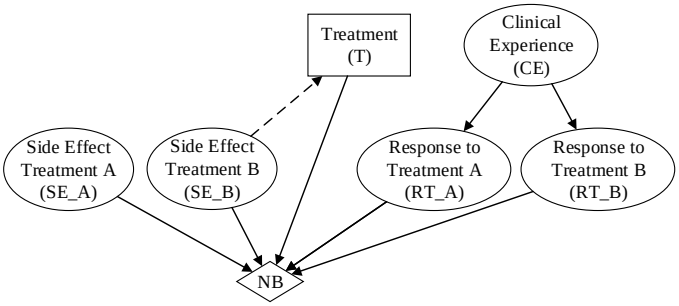


Figure 8 Modified ID for analyzing EVPPI of SE_B

In summary, computation of EVPPI in IDs requires solving two ID models and subtracting their expected utilities. There is an extensive literature on solving discrete IDs. Earlier research on this topic focused on solving IDs by marginalizing variables or transforming them to DTs or BNs [12, 27-30]. Jensen and Dittmer [31] modified the JT algorithm for IDs, which they call a strong JT, and developed a special propagation scheme to compute expected utilities. Dittmer and Jensen [32] proposed a VOI approach to compute EVPPI directly on strong JTs. Shachter [33] focused on improving this approach by generating more efficient strong JTs for VOI, and reusing them for different EVPPI analyses. Liao and Ji [34] proposed an approach that can evaluate the combined EVPPI for a series of observations in IDs with certain constraints. Since most popular BN algorithms, including JT, were designed to solve discrete models, ID algorithms that use BN conversion or strong JT only apply to discrete IDs. Solving HIDs is, however, a more challenging task. Initial research on solving HIDs focused on Gaussian distributions due their convenient computational properties [35-37]. Cobb and Shenoy [38], Cobb and Shenoy [39] and Li and Shenoy [40] proposed a method that can adopt a wider variety of statistical distributions by approximating continuous chance and utility nodes to mixtures of truncated exponential functions and mixtures of polynomials. However, these methods are not closed for non-linear deterministic functions and their computation cannot currently be fully-automated. MCMC methods have also been used to compute HIDs [41, 42] but their limitations are similar to the Markov chain VOI methods discussed in the previous section.

## IV. Computing EVPPI in Hybrid IDs

In this section, we describe a novel method to compute the EVPPI of discrete and continuous variables in HIDs using the dynamic discretization (DD) algorithm. Sections IV.A and IV.B respectively describe the DD algorithm and show how DD is used to solve HIDs. Section IV.C presents a method to compute EVPPI using DD, and Sections IV.D and IV.E show two approximations for the proposed method. Section IV.F illustrates the use of the proposed method.

### *A. Dynamic Discretization Algorithm*

Until relatively recently, the apparent intractability of solving HBNs and HIDs (i.e. BNs and IDs with both discrete and continuous variables) was one of the main limitations of using these modelling approaches for complex decision problems. However, the advent of the DD algorithm [10] now offers a powerful and flexible solution to solve such models. The DD algorithm was developed for propagation in HBN models. Since a HID can be transformed to a HBN (which is discussed in Section IV.B), the DD algorithm also offers a powerful approach for solving HIDs.

The DD algorithm iteratively discretizes the domain of continuous variables in a HBN model by minimizing the relative entropy between the true and the discretized marginal probability densities. It adds more states to high-density areas and merges states in the zero-density areas. At each iteration, DD discretizes each continuous variable in the area of highest density, and then a standard propagation algorithm for discrete BNs, such as the JT algorithm, is used to calculate the posterior marginal given this discretization. The JT algorithm computes

the posteriors of a discrete BN by transforming the BN structure into a tree structure with clusters, which is called a JT. The discretization of all continuous variables in the JT are revised by the DD algorithm every time new evidence is entered into the BN.

The approximate relative entropy error between the true probability density function *f* and its discretization is computed by

$$E_j = \left[\frac{f_{max} - \bar{f}}{f_{max} - f_{min}} f_{min} log \frac{f_{min}}{\bar{f}} + \frac{\bar{f} - f_{min}}{f_{max} - f_{min}} f_{max} log \frac{f_{max}}{\bar{f}}\right] |w_j| \quad (8)$$

where $E_j$ is the approximate relative entropy error, and $f_{max}$, $f_{min}$, $\bar{f}$ are the maximum, minimum and mean values of the function in a given discretization interval $w_j$ respectively.

The convergence threshold of the DD algorithm sets an upper bound relative entropy for stopping the algorithm. The algorithm stops discretizing a node if the sum of approximate entropy errors of all intervals of the node is smaller than the convergence threshold. The relative entropy decreases as the discretization has more states, and it approaches zero as the number of discretized states approaches infinity. Therefore, the user can set the trade-off between the speed of computation and accuracy of the discretization by using the convergence threshold. DD provides an accurate posterior model as the discretization chosen for the marginal is applied to all clusters in the JT. The DD algorithm is formally summarized as follows:

*Choose an initial discretization for all continuous variables in the BN*

*Define the convergence threshold CT and the maximum number of iterations MN*

**for** *each iteration until MN*

*Compute the NPT for each node for the current discretization*

*Enter evidence, and compute propagation in the JT using a standard JT algorithm* [14]

**for** *each continuous node*

*Get the posterior marginal for each node.*

*Compute the approximate relative entropy error between the true and discretized distributions by using Equation 8.*

**if** *the approximate relative entropy error is smaller than CT*

*Stop discretization for this node*

**else**

*Split the interval with the highest entropy error*

*Merge consecutive intervals with zero entropy errors*

**end if**

**end for**

**end for**

For a given threshold, the DD algorithm computes the optimal discretization of any parameterized statistical distribution or conditionally deterministic functions for chance and utility nodes. A fully automated version of the DD algorithm is implemented in AgenaRisk [43]. Readers are referred to [10, 44] for technical details, performance assessments and comparisons of the DD algorithm with other approximation methods.

### *B. Inference in Hybrid IDs using DD*

An algorithm to solve HIDs using DDs has recently been developed by Yet et al.[11]. This approach has two main stages: first a HID is transformed to a HBN, then the DD algorithm is used together with JT to propagate the HBN, and a minimal DT containing only decision and observable chance nodes is generated from the propagated HBN. The optimal decision policies are shown on the minimal DT. The steps of this approach are as follows:

**Step 1 - Transform HID to HBN:**

*Record the sequential order of the decisions* $\bar{D} = D_1, \dots, D_k$ *and observable chance nodes* $\bar{O} = O_1, \dots, O_p$ *according to the information arcs in the HID;*

***for*** *each decision node* $D_i$ *in* $\bar{D}$

*Convert Di to a corresponding BN node* $\Delta_i$

*Convert incoming information arcs of Di to conditional arcs*

***for*** *each state* $d_{ij}$ *of the decision node* $D_i$

*Convert* $d_{ij}$ *to a state* $\delta_{ij}$ *of the corresponding BN node* $\Delta_i$

***if*** *there is asymmetry regarding* $d_{ij}$

*Assign zero probabilities to those state combinations associated with* $\delta_{ij}$

***end if***

*Assign uniform probabilities to the rest of the state combinations*

***end for***

***end for***

*Transform the utility nodes* $\bar{U} = U_1, \dots, U_l$ *into continuous BN nodes* $\bar{Y} = Y_1, \dots, Y_l$

**Step 2 - Propagate the HBN and prepare a minimal DT:**

*Propagate the HBN*

*Call* **PrepareMinimalDT**(*1st node in the sequential order*)

*Evaluate the decision tree using the 'average out and fold back' algorithm as described in Korb and Nicholson [45].*

The function PrepareMinimalDT is defined as follows:

**PrepareMinimalDT**(*ith node in the sequential order*)

**for** *each state of the node:*

*Remove all evidence from the node and subsequent nodes in the sequential order.*

**if** *a corresponding node does not exist in the DT*

**if** *i* = 1

*Add a decision or chance node to the DT corresponding to the type of the node in the HID.*
**else**
*Add a decision or chance node next to the last arc added in the DT corresponding to the type of the node in the HID.*
**end if**
**end if**
*Add an arc next to the corresponding node in the DT. Label the name of the state on that arc.*
**if** *the current state is from an observable chance node*
*Label its posterior probability from the HBN on the arc added in the DT.*
**end if**
*Instantiate the state and propagate the HBN.*
**if** *the state entered is from the last node in the sequential order*
*Add a utility node next to the last arc added in the DT, and label the value of this node with the posterior value of the aggregate utility node from the HBN.*
**else**
*Recursively call* **PrepareMinimalDT** *by using the* $i$+1$^{th}$ *node in the sequential order.*
**end if**
**end for**

This approach allows all commonly used statistical distributions, and any commonly used linear or non-linear conditionally deterministic function of those distributions, to be used for chance and utility nodes. It presents the computed decision strategies in a minimal DT that shows only the decision and observable chance nodes. In the following section, we use this algorithm to compute EVPPI in HIDs.

### C. Computing EVPPI using DD

The steps for computing the EVPPI of any discrete or continuous unobservable chance variable $X \in \overline{N}$ before decision $D \in \overline{D}$ in an ID $G$ are:

1. *Build the modified ID G' by adding an information arc from X to D;*
2. *Solve G and G' by using the algorithm described in Section IV.B;*
3. *EVPPI(X) = EU(G') – EU(G).*

The complexity of the JT algorithm is exponential in the largest cluster in the JT. Our algorithm transforms an HID to a HBN and solves it by using DD and JT algorithms. Calculating G and G' can be computationally expensive especially if there are multiple decision and observable chance nodes with many states. In the following section we present a further approximation of the proposed EVPPI technique that enables us to trade-off accuracy with speed.

### D. Further Approximation of EVPPI using DD

The algorithm described in Section IV.B enters evidence for each state combination of the observable chance and decision nodes. The DD algorithm revises its discretizations every time evidence is entered. To calculate EVPPI of an unobservable chance variable, it needs to be transformed into an observable variable in the modified ID as described in Section IV.C. Therefore, the modified ID has an even higher number of state combinations of observable chance and decision nodes. Rather than revising the discretizations, a further approximation of our EVPPI technique uses DD only once initially, and then generates a fixed discretization of this model. The propagations, for all state combinations, in the following steps are applied to this fixed discretization by using only JT. This means observations entered in to the model preserve the prior discretized points on all unobserved variables. We call this approximation DD with Fixed discretization (DD-Fixed).

As we show later in Section V, DD-Fixed works faster but is also less accurate than DD because the discretizations are not optimized for the posteriors in each iteration. In the following section, we show another approximation that computes only the expected utility and EVPPI values rather than the whole distribution of them.

### E. External Calculation of Utility Functions

In health economics models, the utility function is a deterministic function of random variables such as costs and life expectancy. The DD algorithm can solve hybrid models with all common conditional deterministic functions of random variables. It computes the entire probability distribution of such functions and thus it enables the decision maker to assess the uncertainty of the model's estimates as well as the expected values.

In an HID, deterministic utility functions are modelled as a child node to all the variables of that function. Therefore, this node has many parents if the utility function has many variables. The computational complexity of standard propagation algorithms, such as JT, depends on the largest clique size, and the clique size can explode if a variable has many parents. Although solutions have been proposed for this [46], solving a HID that has a utility node with many parents can still be slow or infeasible.

If a decision maker is interested in the expected value of additional information, computing the expected values of the utility function, rather than the entire probability distribution, is sufficient. In this case, we can calculate the posteriors of the marginal probability distributions of the independent variables and joint probability distribution of the dependent variables in the utility function, and then we can simply sum and multiply the expected values of the variables externally. For example, in Figure 8, the utility functions associated with each treatment alternative are '1000×*RT_A* – 4000×*SE_A"* and '1000×*RT_B* – 4000×*SE_B*' respectively. Since *RT_A* and *SE_A*, and *RT_B* and *SE_B* are respectively independent of each other, we can simply calculate their posterior distribution, and then apply the utility functions to the expected values obtained from these variables. We only calculate the expected values of *NB*, rather

than the whole distribution, resulting in much faster computation of EVPPI.

### *F. EVPPI Example*

In this section, we illustrate the use of the proposed EVPPI technique based on the HID models shown in Figure 6 and Figure 7. We compute the EVPPI of a discrete node (*CE*) and a continuous node (*SE_A*). Note that these HIDs model exactly the same problem as discussed in Section II.B.3.

#### *1) EVPPI of CE*

We first create the modified ID *G'* by adding an information arc from *CE* to *T*. Then we solve *G* and *G'* by using the HID solver algorithm described in Section IV.C. This algorithm first converts the IDs to BNs as described below:

1. Sequential order of decisions and observations is recorded based on the information arcs. The original ID *G* has no sequential order as it has only one decision and no information arcs. The modified ID *G'* has an information arc between *CE* and *T*, thus it has the sequential order $CE \prec T$, i.e. *CE* is observed before decision *T* is made.
2. The decision nodes and information arcs are transformed into BN nodes and conditional arcs respectively. Since there is no asymmetry in these models, the decision nodes have uniform distributions.
3. The utility node is modelled as a mixture distribution conditioned on the decision node.

After both *G* and *G'* are converted to BNs, the algorithm instantiates and propagates the BNs for all possible state combinations in the sequential order. The original ID, *G*, has only one decision node, *T*, in the sequential order. The minimal DT for *G* is built by propagating the BN for each decision alternative and recording the expected values of the posterior utilities as shown in Figure 9. The optimal decision is treatment *B* with an expected utility of 19.03.

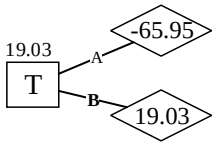


Figure 9 Minimal DT from Original ID

The sequential order of the decision and observable chance nodes of the modified ID *G'* is $CE \prec T$. Next, the minimal DT is built by using the algorithm described in Section IV.B. The steps of this algorithm are as follows:

1. Starts generating the minimal DT with the first node in the sequential order, i.e. *CE*. The DT is initially empty, therefore it adds a chance node labelled 'CE'.
2. Adds an arc next to this node for its first state *'CE = Trainee'* and labels it with the name and probability of this state.
3. Instantiates *'CE = Trainee'* and propagates the BN model.
4. The next node in the sequence is *T*. Adds a decision node labelled *'T'* in the DT.
5. Adds an outgoing arc from this node and labels it with its first state 'A'. Since *T* is a decision node, a probability is not labelled on this arc.
6. Instantiates '*T = A*' and propagates the BN model.
7. Since '*T = A*' is the last node in the sequential order of this state combination, it adds a utility node next this node in the DT, and labels it with the posterior of the utility node.
8. Clears the evidence entered on '*T'* and continues with its second state '*T = B'*. It adds an arc labelled 'B' next to the node 'T'.
9. Instantiates '*T = B'*, propagates the BN, and adds another utility node in the DT.
10. Since the algorithm has evaluated all states of '*T*', it continues the second state of *'CE'.*

The algorithm analyzes the remainder of the state combinations in the same way as above by using the algorithm described in Section IV.C.

Figure 10 shows the minimal DT built from the modified BN. The optimal decision policy is selecting treatment *A* when the treatment is applied by a trainee and selecting treatment B when it is applied by an experienced clinician. The expected utility of this policy is 38.58.

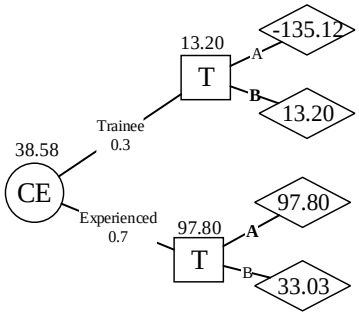


Figure 10 Minimal DT for the modified BN

The EVPPI of CE is the difference between the expected utility values of optimal decisions in each these graphs:

$$EVPPI(CE) = 38.58 - 19.03 = 19.55 \quad (9)$$

#### *2) EVPPI of SE_A*

In order to compute the EVPPI of *SE_A* in Figure 7, we also add an information arc from this node to the decision node (see Figure 8). The DD algorithm provides an optimal discretization of *SE_A* for the given convergence threshold, and therefore enables us to solve this HID and compute the EVPI of *SE_A* as if it were a discrete node. Figure 11 shows a discretization of *SE_A* by using DD with a convergence threshold of $10^{-3}$.

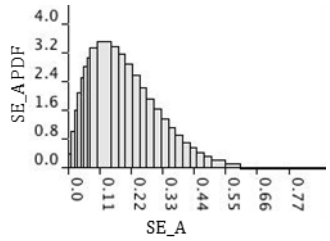


Figure 11 Dynamic Discretization of *SE_A*

Table 2 States, Probabilities and Expected Utilities of *SE_A* computed by DD

| ***SE_A*** | **T** | **P(*SE_A*)** | **NB(*T, SE_A*)** |
|---|---|---|---|
| **[0, 0.01)** | A | 0.00340 | 713.93 |
| **…** | | … | |
| **[0.775, 1]** | A | 0.00005 | -2881.44 |
| **[0, 0.01]** | B | 0.00340 | 19.03 |
| **…** | | | |
| **[0.775, 1]** | B | 0.00005 | 19.03 |

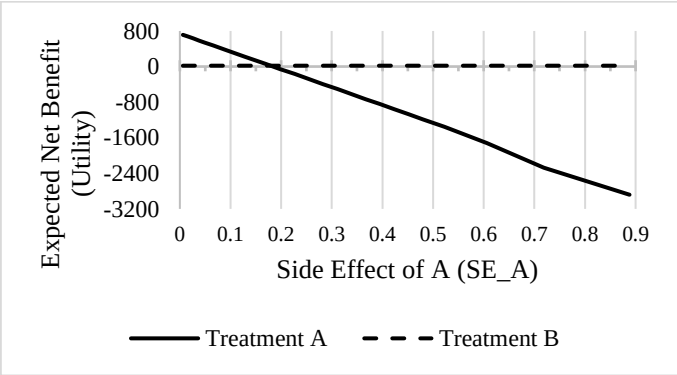


Figure 12 Expected Net Benefit of Treatments A and B given Perfect Information on SE_A

Table 2 shows a subset of the discretized states and probability values of *SE_A* together with the state of the associated decision variable in the state combination, and the expected utility value. For example, when *SE_A* is between 0 and 0.01, and the treatment decision is *A*, the expected utility of this combination is 713.93, and $P(0 \leq SE\_A < 0.01) = 0.0034$. We use these values to build the minimal DT and compute the optimal decision policy. However, since discretized continuous variables often have a large number of states, building a DT for all of these state combinations would have many branches with the same decision policy. Rather than showing each discretized state of the continuous chance nodes, we can show its intervals where the optimal decision policy is the same. Figure 12 shows the expected utilities of treatments A and B given different states of SE_A. The optimal decision policy is $T = A$ for all states of *SE_A* between [0,0.184) because the expected utility of $T = A$ is more than $T = B$ for these states. Therefore, we can combine the branches associated with these states in the DT to get a simpler and clearer DT. Figure 13 shows the minimal DT for computing the expected utility of the modified ID.

The EVPPI of *SE_A* is:

$$EVPPI(SE_A) = 167.84 - 19.03 = 148.81 \qquad (10)$$

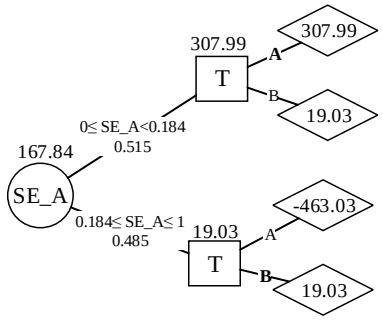


Figure 13 Minimal DT of the modified BN for SE_A

## V. Case Study

In this section, we compare the results of the proposed EVPPI technique and its approximations (i.e. DD and DD-Fixed) to the nested two-level sampling approach [2], the GAM and GP regression approaches [4], and to another two approximate EVPPI techniques proposed by Strong and Oakley [3], Sadatsafavi, et al. [5] (see Section III.A for a discussion of these techniques). The nested two-level sampling, Strong and Oakley's and Sadatsafavi et al.'s techniques will be referred as NTL, STR and SAD respectively.

Table 3 shows the sampling settings used for these techniques. Since the results of sampling-based approaches differ slightly every time they are computed, we repeated their analyses 100 times and present the average of their results.

We assumed that the "true" EVPPI value of each parameter is the average result of 100 NTL analyses, with 10000 × 1000 inner and outer level samples, for that parameter,

Table 3 Sampling Based EVPPI Approaches used in Case Study

| **Approach** | **Sampling Settings** |
|---|---|
| **NTL [2]** | 10000 outer level, 1000 inner level samples |
| **GAM [4]** | 10,000 samples |
| **GP [4]** | 1000 samples |
| **STR [3]** | 50 blocks, 10,000 samples |
| **SAD [5]** | 1 separator, 10,000 samples |

AgenaRisk and was used to compute DD and DD-Fixed. JAGS and R software were used together with R2JAGS interface to compute EVPPIs with NTL. The BCEA package of R was used in addition to JAGS and R2JAGS to compute EVPPIs using GAM, GP, STR and SAD. The remainder of this section shows the results of these methods in calculating EVPPI for two case studies.

### A. Case Study 1

For the first case study, we have used the model described in Chapter 3 of Baio [47]. This model calculates the cost-effectiveness of two alternative drugs. A description of the model structure and parameters is shown in the appendix. An equivalent ID structure for this model is shown in Figure 14. We calculated the EVPPI for the parameters ρ, γ and π[1] as the rest of the unobservable parameters are defined based on these variables by simple arithmetic expressions or binomial distributions.

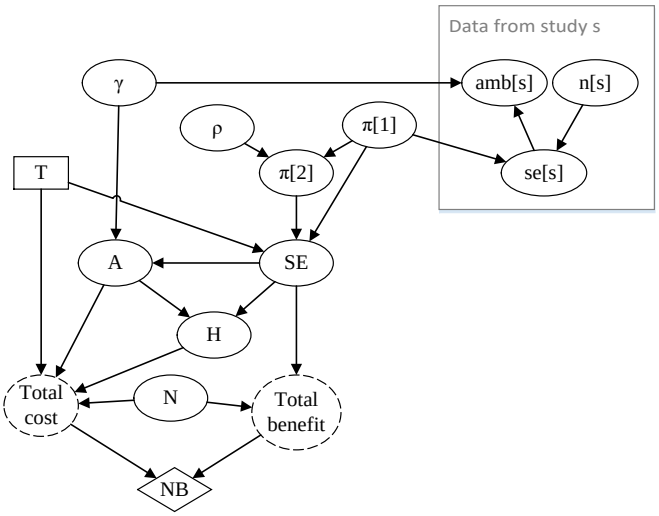


Figure 14 ID model – Case Study 1

*1) DD and DD-Fixed with Different Convergence Thresholds*

We tested DD and DD-Fixed with 5 different convergence thresholds settings: i.e. $10^{-1}$, $10^{-2}$, $10^{-3}$, $10^{-4}$, and $10^{-5}$. Figure 15 shows the EVPPI values of ρ and the calculation times when different thresholds settings are used. The horizontal line shown in Figure 15a shows the "true" NTL result.

Both the DD and DD-Fixed approaches provide an accurate approximation of the EVPPI value even with high convergence thresholds. DD accurately computes the EVPPI starting from the convergence threshold of $10^{-2}$. Its calculation time is 19 and 650 seconds at $10^{-2}$ and $10^{-3}$ thresholds respectively. The calculation time of DD increases exponentially after the convergence threshold of $10^{-3}$ but the EVPPI results do not change.

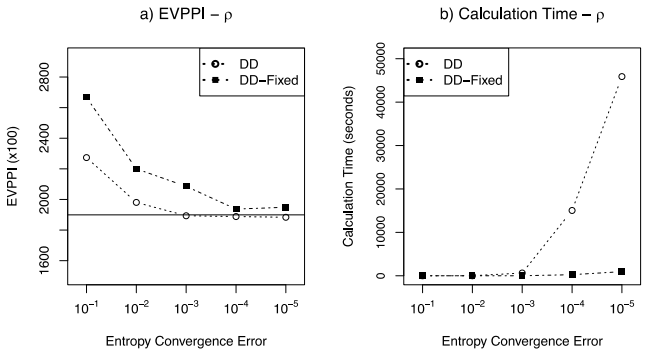


Figure 16 EVPPI Values and Calculation Time of ρ

The results of DD-Fixed is close to the true value when the convergence threshold is $10^{-4}$ and $10^{-5}$. Its calculation time is 14 and 266 seconds at $10^{-3}$ and $10^{-4}$ thresholds respectively. DD-Fixed is considerably faster than DD but it does not compute the EVPPI as accurately as DD at any convergence threshold setting. This is expected, as DD calculates the posteriors more accurately by revising discretizations at every step. The EVPPI of γ and π[1] are 0. DD is able to find the correct value at all convergence thresholds used. DD-Fixed calculates a positive EVPPI for π[1] at $10^{-1}$ thresholds, and is able to find the correct value starting from $10^{-2}$.

*2) Comparison with Other Approaches*

Table 4 shows the results of DD, DD-Fixed and NTL in the first case study. The results of DD and DD-Fixed are calculated with the convergence thresholds of $10^{-3}$ and $10^{-4}$ respectively. We selected these threshold settings because smaller thresholds have a much higher calculation time without any substantial accuracy benefit. A convergence threshold setting of $10^{-2}$ for DD also provides similar EVPPIs much faster as shown in the previous section.

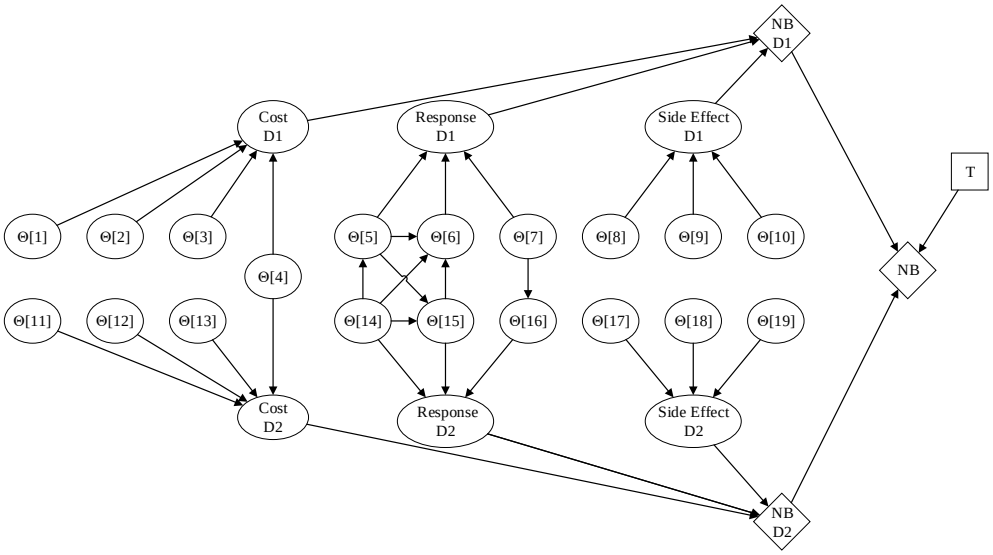


Figure 15 ID Structure – Case Study 2

Table 4 EVPPI using DD, DD-Fixed and NTL in Case Study 1

| Parameter | DD | Time (sec) | DD-Fixed | Time (sec) | NTL | Time (sec) |
|---|---|---|---|---|---|---|
| **ρ** | 189,329 | 650 | 193,682 | 266 | 189,333 | 1756 |
| **γ** | 0 | 74 | 0 | 146 | 0 | 1608 |
| **π[1]** | 0 | 326 | 0 | 155 | 0 | 1603 |

The EVPPIs of ρ calculated by DD and NTL are very close, and γ and π[1] have no value of information. DD-Fixed finds a slightly higher value for the EVPPI of ρ, and its calculation is faster.

Table 5 EVPPI using GAM, GP, STR and SAD in Case Study 1

| Parameter | GAM | Time (sec) | GP | Time (sec) |
|---|---|---|---|---|
| **ρ** | 188,141 | 12 | 190,689 | 11 |
| **γ** | 0 | 12 | 1.6 | 6 |
| **π[1]** | 0 | 12 | 15.2 | 10 |
| **Parameter** | **STR** | **Time (sec)** | **SAD** | **Time (sec)** |
| **ρ** | 188,133 | 6 | 188,590 | 6 |
| **γ** | 0 | 6 | 136.6 | 6 |
| **π[1]** | 0 | 6 | 142.1 | 6 |

GAM, GP, STR and SAD find similar values for ρ with only 1% - 3% difference from the DD, DD-Fixed and NTL results (see Table 5). SAD, however, finds positive values for the EVPPI of γ and π[1] while these values are supposed to be 0.

### *B. Case Study 2*

We used the model described by Brennan, et al. [2] for our second case study. The structure and parameters of this model are shown in the appendix, and an equivalent ID structure for this model is shown in Figure 15. The parameters Θ[5], Θ[7], Θ[14] and Θ[16] have a multivariate normal distribution with a pairwise correlation coefficient of 0.6. Similarly, the parameters Θ[6] and Θ[15] have a bivariate normal distribution and the same pairwise correlation coefficient. Since the parameters of each variable in an ID represent a CPD, the multivariate Normal distributions are modelled as multiple CPDs in the ID [48]. We calculated the EVPPI for these 6 parameters as the rest of the parameters in this model are independent of each other and have very low or no value of information.

#### *1) DD and DD-Fixed with Different Convergence Thresholds*

Figures 17 and 18 show the EVPPI values and the calculation times of Θ[5], Θ[6], Θ[7], Θ[14] , Θ[15] and Θ[16] when different convergence threshold settings are used. The horizontal lines show the NTL results. Both DD and DD-Fixed accurately calculate the EVPPI values of all parameters at the convergence threshold of $10^{-3}$.

In the DD approach, calculation of the EVPPI for both Θ[6] and Θ[16] took significantly longer than the other parameters. This is possibly caused by the way multivariate Gaussian distributions are modelled in the ID model. These variables have several other parents, and this increases their calculation time in DD and JT. There is ongoing research to speed up inference of such variables with many parents by using region based approximations.

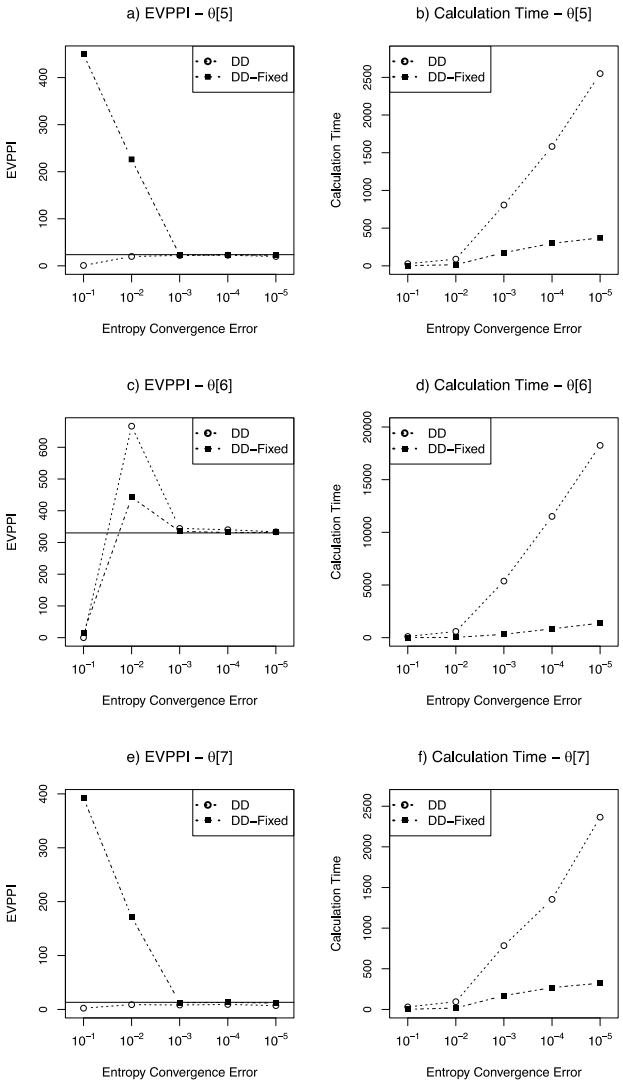


Figure 17 EVPPI and Calculation Times of Θ[5], Θ[6] and Θ[7]

#### *2) Comparison with Other Techniques*

Table 6 shows the results of DD, DD-Fixed and NTL in the second case study, and Table 7 shows the results of the approximate sampling-based approaches. DD and DD-Fixed were calculated with a convergence threshold setting of $10^{-3}$ and $10^{-4}$ respectively. The results of all approaches are close to each other and to the results of NTL. The calculation time of DD is higher than the alternative approaches especially in cases where a variable has many parents. DD-Fixed is faster than DD and NTL, but still slower than GAM, GP, STR and SAD.

Table 6 EVPPI using DD, DD-Fixed and NTL in Case Study 2

| Parameter | DD | Time (sec) | DD-Fixed | Time (sec) | NTL | Time (sec) |
|---|---|---|---|---|---|---|
| **Θ[5]** | 22.06 | 807 | 23.45 | 298 | 23.73 | 1725 |
| **Θ[6]** | 344.157 | 5373 | 331.48 | 835 | 329.98 | 1586 |
| **Θ[7]** | 8.223 | 785 | 12.95 | 269 | 13.13 | 1573 |
| **Θ[14]** | 230.55 | 1009 | 231.04 | 346 | 232.02 | 1589 |
| **Θ[15]** | 0 | 752 | 0 | 329 | 0 | 1583 |
| **Θ[16]** | 465.86 | 4410 | 466.22 | 778 | 461.47 | 1657 |

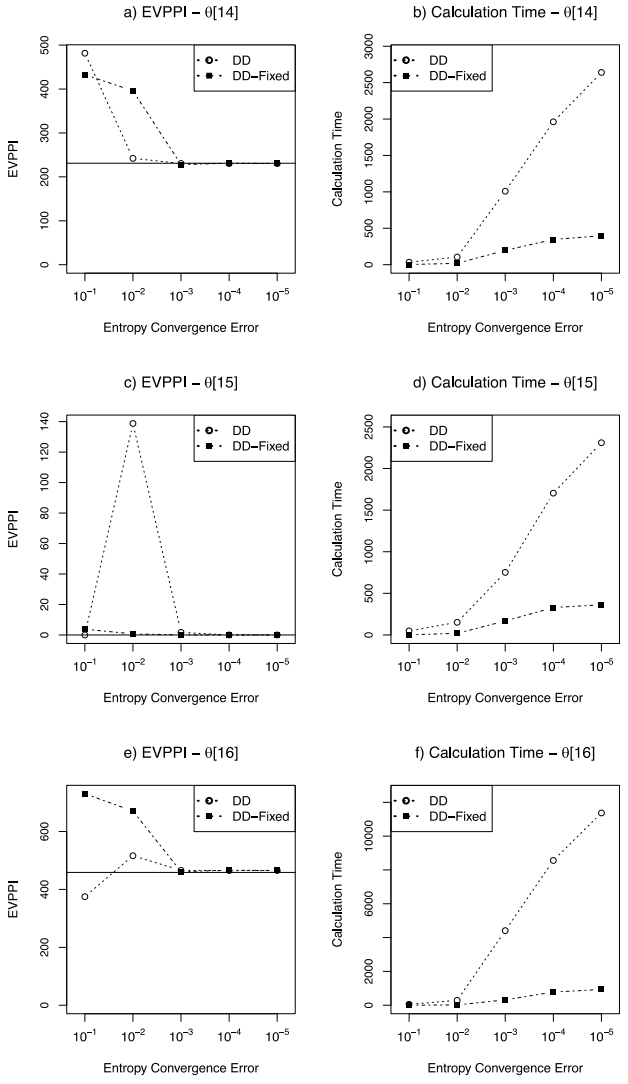


Figure 18 EVPPI and Calculation Times of Θ[14], Θ[15] and Θ[16]

### C. Summary of the Results

Both the DD and DD-Fixed approaches accurately calculate the EVPPI values even with high convergence threshold settings. Starting from the convergence threshold of $10^{-2}$, the results of DD are close to the results of the NTL approach with 10,000 × 1000 inner and outer level samples, which we assume to be the "true" result. The results of both DD-Fixed and DD converge to the true values starting from the $10^{-3}$ threshold setting. This also illustrates that DD successfully calculates the posteriors of the utility variables that are composed of mixture distributions with constants.

Table 7 EVPPI using GAM, GP, STR and SAD in Case Study 2

| Parameter | GAM | Time (sec) | GP | Time (sec) |
|---|---|---|---|---|
| **Θ[5]** | 23.92 | 7 | 26.24 | 6 |
| **Θ[6]** | 329.27 | 7 | 341.62 | 6 |
| **Θ[7]** | 14.31 | 7 | 17.42 | 6 |
| **Θ[14]** | 231.31 | 7 | 234.03 | 6 |
| **Θ[15]** | 0.01 | 7 | 0.46 | 6 |
| **Θ[16]** | 458.82 | 7 | 463.20 | 6 |
| **Parameter** | **STR** | **Time (sec)** | **SAD** | **Time (sec)** |
| **Θ[5]** | 23.74 | 5 | 26.54 | 5 |
| **Θ[6]** | 329.70 | 5 | 333.11 | 5 |
| **Θ[7]** | 13.96 | 5 | 16.70 | 5 |
| **Θ[14]** | 232.46 | 5 | 235.54 | 5 |
| **Θ[15]** | 0 | 5 | 0.02 | 5 |
| **Θ[16]** | 458.95 | 5 | 461.70 | 5 |

Computation speed was the main limitation of our method. Although both DD and DD-Fixed were generally faster than NTL, they were slower than the sampling-based approaches at all convergence threshold settings with acceptable accuracy.

However, the main advantage of our method is its usability as it is based on discretized posterior distributions that are automatically handled by the DD algorithm which is implemented in a widely available tool. Therefore, in contrast to the other approaches, it does not require users to assess the convergence of sampling.

## VI. Conclusions

This paper presented a novel technique to calculate the EVPPI of individual continuous variables in IDs, and an approximation of this technique to trade-off accuracy for speed. We demonstrated the use of this technique and applied it to two case studies that were used in similar studies. We compared the results of our approach to five other general techniques used for computing EVPPI, namely those of Brennan, et al. [2], Strong and Oakley [3], Sadatsafavi, et al. [5], and the GP and GAM regression methods of Strong, et al. [4]. While all previous techniques use sampling to calculate EVPPI, our approach uses an entirely different technique that dynamically discretizes all continuous variables, and calculates the posteriors by using a popular Bayesian inference algorithm called JT. As a result, it can handle a large class of models with virtually any kind of continuous distribution. Our approach successfully calculated EVPPIs for individual variables in both case studies. In contrast to the previous techniques, our approach can be used by decision-makers who do not have deep understanding of programming languages and sampling techniques, since it offers a simpler way of calculating EVPPI. The case studies show that, while our approach requires longer computation times, there is no compromise on EVPPI accuracy.

Our technique uses a powerful inference algorithm that is readily implemented in a commercial tool with a user-friendly graphical interface. Application of the technique requires only simple graphical operations on the ID models and computing these models by using the proposed algorithms. As further

research, the proposed approach could be extended to calculate the EVPPI of a group of parameters. This could be achieved by adding multiple information arcs on an ID and computing the difference between expected utilities. An automated implementation of the EVPPI algorithm and the proposed approximations would enable a wider use of these techniques by clinicians and domain experts. We also plan to further evaluate the general accuracy of DD-Fixed approximation in HBN and HID models.

## REFERENCES


[1] C. Minelli and G. Baio, "Value of information: a tool to improve research prioritization and reduce waste," *PLoS medicine*, vol. 12, no. 9, p. e1001882, 2015.

[2] A. Brennan, S. Kharroubi, A. O'Hagan, and J. Chilcott, "Calculating Partial Expected Value of Perfect Information via Monte Carlo Sampling Algorithms," *Medical Decision Making*, vol. 27, no. 4, pp. 448-470, July 1, 2007 2007.

[3] M. Strong and J. E. Oakley, "An Efficient Method for Computing Single-Parameter Partial Expected Value of Perfect Information," *Medical Decision Making*, vol. 33, no. 6, pp. 755-766, August 1, 2013 2013.

[4] M. Strong, J. E. Oakley, and A. Brennan, "Estimating Multiparameter Partial Expected Value of Perfect Information from a Probabilistic Sensitivity Analysis Sample: A Nonparametric Regression Approach," *Medical Decision Making*, vol. 34, no. 3, pp. 311-326, April 1, 2014 2014.

[5] M. Sadatsafavi, N. Bansback, Z. Zafari, M. Najafzadeh, and C. Marra, "Need for Speed: An Efficient Algorithm for Calculation of Single-Parameter Expected Value of Partial Perfect Information," *Value in Health*, vol. 16, no. 2, pp. 438-448, 3// 2013.

[6] J. Madan *et al.*, "Strategies for Efficient Computation of the Expected Value of Partial Perfect Information," *Medical Decision Making*, January 21, 2014 2014.

[7] M. Strong, J. E. Oakley, A. Brennan, and P. Breeze, "Estimating the expected value of sample information using the probabilistic sensitivity analysis sample: a fast, nonparametric regression-based method," *Medical Decision Making*, vol. 35, no. 5, pp. 570-583, 2015.

[8] A. Heath, I. Manolopoulou, and G. Baio, "A Review of Methods for Analysis of the Expected Value of Information," *Medical decision making*, 2017.

[9] R. A. Howard and J. E. Matheson, "Influence diagrams," *Decision Analysis*, vol. 2, no. 3, pp. 127-143, 2005.

[10] M. Neil, M. Tailor, and D. Marquez, "Inference in hybrid Bayesian networks using dynamic discretization," (in English), *Statistics and Computing*, vol. 17, no. 3, pp. 219-233, 2007/09/01 2007.

[11] B. Yet, M. Neil, N. Fenton, A. Constantinou, and E. Dementiev, "An Improved Method for Solving Hybrid Influence Diagrams," *to appear in International Journal of Approximate Reasoning*.

[12] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. San Francisco, CA: Morgan Kaufmann, 1988.

[13] N. Fenton and M. Neil, *Risk assessment and decision analysis with Bayesian networks*. CRC Press, 2012.

[14] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 50, no. 2, pp. 157-224, 1988.

[15] F. V. Jensen and T. D. Nielsen, *Bayesian networks and decision graphs*. Springer, 2009.

[16] R. D. Shachter, "Probabilistic inference and influence diagrams," *Operations Research*, vol. 36, no. 4, pp. 589-604, 1988.

[17] J. A. Tatman and R. D. Shachter, "Dynamic programming and influence diagrams," *IEEE transactions on systems, man, and cybernetics*, vol. 20, no. 2, pp. 365-379, 1990.

[18] K. P. Murphy, "Dynamic bayesian networks: representation, inference and learning," PhD, University of California, Berkeley, 2002.

[19] H. C. Sox, M. C. Higgins, and D. K. Owens, *Medical decision making*, 2 ed. West Sussex, UK: Wiley, 2013.

[20] M. M. Hunink *et al.*, *Decision making in health and medicine: integrating evidence and values*. Cambridge University Press, 2014.

[21] E. Fenwick, S. Palmer, K. Claxton, M. Sculpher, K. Abrams, and A. Sutton, "An iterative Bayesian approach to health technology assessment: application to a policy of preoperative optimization for patients undergoing major elective surgery," *Medical Decision Making*, vol. 26, no. 5, pp. 480-496, 2006.

[22] J. E. Oakley, A. Brennan, P. Tappenden, and J. Chilcott, "Simulation sample sizes for Monte Carlo partial EVPI calculations," *Journal of health economics*, vol. 29, no. 3, pp. 468-477, 2010.

[23] G. Baio, A. Berardi, and A. Heath, "Bayesian cost effectiveness analysis with the R package BCEA," ed: Springer, 2016.

[24] G. Baio, P. Hadjipanayiotou, A. Berardi, and A. Heath. (2017). *BCEAWeb*. Available: https://egon.stats.ucl.ac.uk/projects/BCEAweb/

[25] M. Strong, J. Oakley, and A. Brennan. (2016, 25/07/2017). *SAVI - Sheffield Accelerated Value of Information*. Available: http://savi.shef.ac.uk/SAVI/

[26] H. Jalal and F. Alarid-Escudero, "A Gaussian Approximation Approach for Value of Information Analysis," *Medical Decision Making*, vol. 0, no. 0, p. 0272989X17715627.

[27] R. D. Shachter, "Evaluating influence diagrams," *Operations research*, vol. 34, no. 6, pp. 871-882, 1986.

[28] G. F. Cooper, "A Method for Using Belief Networks as Influence Diagrams," in *Fourth International Conference on Uncertainty in Artificial Intelligence*, Minneapolis, Minnesota, USA, 1988, pp. 55-63: North-Holland.

[29] R. D. Shachter and M. A. Peot, "Decision making using probabilistic inference methods," in *Proceedings of the Eighth international conference on Uncertainty in artificial intelligence*, 1992, pp. 276-283: Morgan Kaufmann Publishers Inc.

[30] N. L. Zhang, "Probabilistic inference in influence diagrams," *Computational Intelligence*, vol. 14, no. 4, pp. 475-497, 1998.

[31] F. V. Jensen and S. L. Dittmer, "From influence diagrams to junction trees," in *Proceedings of the Tenth international conference on Uncertainty in artificial intelligence*, 1994, pp. 367-373: Morgan Kaufmann Publishers Inc.

[32] S. L. Dittmer and F. Jensen, "Myopic Value of Information in Influence Diagrams," *Uncertainty in Artificial Intelligence: Proceedings of the Thirteenth Conference*, pp. 142-149, 1997.

[33] R. D. Shachter, "Efficient value of information computation," *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 594-601, 1999.

[34] W. Liao and Q. Ji, "Efficient non-myopic value-of-information computation for influence diagrams," *International Journal of Approximate Reasoning*, vol. 49, pp. 436-450, 2008.

[35] W. B. Poland III, "Decision analysis with continuous and discrete variables: A mixture distribution approach," PhD Thesis, Stanford University, Stanford, CA, 1994.

[36] A. L. Madsen and F. V. Jensen, "Solving linear-quadratic conditional Gaussian influence diagrams," *International Journal of Approximate Reasoning*, vol. 38, no. 3, pp. 263-282, 2005.

[37] R. D. Shachter and C. R. Kenley, "Gaussian influence diagrams," *Management science*, vol. 35, no. 5, pp. 527-550, 1989.

[38] B. R. Cobb and P. P. Shenoy, "Inference in hybrid Bayesian networks with mixtures of truncated exponentials," *International Journal of Approximate Reasoning*, vol. 41, no. 3, pp. 257-286, 2006.

[39] B. R. Cobb and P. P. Shenoy, "Decision making with hybrid influence diagrams using mixtures of truncated exponentials," *European Journal of Operational Research*, vol. 186, no. 1, pp. 261-275, 2008.

[40] Y. Li and P. P. Shenoy, "A framework for solving hybrid influence diagrams containing deterministic conditional distributions," *Decision Analysis*, vol. 9, no. 1, pp. 55-75, 2012.

[41] C. Bielza, P. Müller, and D. R. Insua, "Decision analysis by augmented probability simulation," *Management Science*, vol. 45, no. 7, pp. 995-1007, 1999.

[42] J. M. Charnes and P. P. Shenoy, "Multistage Monte Carlo method for solving influence diagrams using local computation," *Management Science*, vol. 50, no. 3, pp. 405-418, 2004.

[43] AgenaRisk. (2018, 23.08.2018). *AgenaRisk: Bayesian network and simulation software for risk analysis and decision support*. Available: http://www.agenarisk.com

[44] P. Lin, M. Neil, and N. Fenton, "Risk aggregation in the presence of discrete causally connected random variables," *Annals of Actuarial Science*, vol. 8, no. 02, pp. 298-319, 2014.

[45] K. B. Korb and A. E. Nicholson, *Bayesian artificial intelligence*. Boca Raton, FL: CRC press, 2010.

[46] M. Neil, X. Chen, and N. Fenton, "Optimizing the calculation of conditional probability tables in hybrid Bayesian networks using binary factorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 7, pp. 1306-1312, 2012.

[47] G. Baio, *Bayesian methods in health economics*. CRC Press, 2012.

[48] P. Lin, "Performing Bayesian Risk Aggregation using Discrete Approximation Algorithms with Graph Factorization," PhD, School of Electronic Engineering and Computer Science, Queen Mary University of London, London, 2014.